

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:  
Laschkewitsch et al.

Serial No.: 09/918,746

Filed: JULY 31, 2001

For: MANAGING INTENDED  
GROUP MEMBERSHIP USING  
DOMAINS

§  
§  
§  
§  
§  
§  
§  
§  
§  
§

Confirmation No.: 2727

Group Art Unit: 2153

Examiner: Philip J. Chea

MAIL STOP APPEAL BRIEF - PATENTS  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**CERTIFICATE OF MAILING OR TRANSMISSION**

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted to the U.S. Patent and Trademark Office to fax number 571-273-8300 to the attention of Examiner Philip J. Chea, on the date shown below:

November 28, 2006  
Date

/Randol W. Read, Reg. No. 43,876/  
Randol W. Read

Dear Sir:

**APPEAL BRIEF**

Applicants submit this Appeal Brief to the Board of Patent Appeals and Interferences on appeal from the decision of the Examiner of Group Art Unit 2153 dated July 25, 2006, finally rejecting claims 8-23. The final rejection of claims 8-23 is appealed. Please charge the amount of \$620.00, the fee of \$500 for filing this brief and the fee of \$120.00 for the one month extension of time, to Deposit Account No. 20-0782/IBM/K10041/RWR.

## **TABLE OF CONTENTS**

1.	Identification Page.....	1
2.	Table of Contents .....	2
3.	Real Party in Interest .....	3
4.	Related Appeals and Interferences .....	4
5.	Status of Claims .....	5
6.	Status of Amendments .....	6
7.	Summary of Claimed Subject Matter .....	7
8.	Grounds of Rejection to be Reviewed on Appeal .....	9
9.	Arguments .....	10
10.	Conclusion .....	14
11.	Claims Appendix .....	15
12.	Evidence Appendix .....	18
13.	Related Proceedings Appendix .....	19

### **Real Party in Interest**

The present application has been assigned to International Business Machines Corporation, Armonk, New York.

### **Related Appeals and Interferences**

Applicant asserts that no other appeals or interferences are known to the Applicant, the Applicant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

### **Status of Claims**

Claims 8-23 are pending in the application. Claims 1-23 were originally presented in the application. Claims 1-7 have been canceled without prejudice. Claims 8-23 stand finally rejected as discussed below. The final rejections of claims 8-23 are appealed. The pending claims are shown in the attached Claims Appendix.

### **Status of Amendments**

All claim amendments have been entered by the Examiner. No amendments to the claims were proposed after the final rejection.

## Summary of Claimed Subject Matter

Claimed embodiments of the invention provide for methods, systems, and articles of manufacture for creating and managing membership of a group of jobs within a cluster by providing a set of interfaces that allows a cluster and a group to be created, and members to be added, removed or joined (see, Paragraph [0026]).

### A. CLAIM 8 - INDEPENDENT

Claim 8 is directed to a method of managing membership of jobs in a cluster (see, e.g., Figure 2; paragraph [0040]). The method includes two parts. First, upon receiving a request to create a group comprising at least two jobs, creating, on a respective node on which each respective job is running, a respective list indicating each of the at least two jobs (see paragraph [0046-0047]). Secondly, upon receiving a request to join the group from a requesting member job having membership to the group, accessing each respective list of each job of the group to determine whether the requesting member job is included in the respective list (see Figure 5; paragraph [0048-0049]).

### B. CLAIM 12 - INDEPENDENT

Claim 12 is directed to a computer system including a plurality of nodes (see Figure 2; paragraph [0038]). Each node includes a processor configured to execute at least one job (see Figure 3 –item 312; paragraph [0042]) and a memory device containing a copy of a first list (see Figure 2 –item 210; paragraph [0038]; Figure 3 –item 334; paragraph [0046]). Each copy of the first list indicates jobs with a membership to a first group (see Figure 2 –item 210A-D; paragraph [0038]) and each job is configured to access its respective copy of the first list to determine whether a requesting job of another node may be joined to the first group (see Figure 5; paragraph [0049]).

### D. CLAIMS 19 - INDEPENDENT

Claim 19 is directed to a memory of a node in a cluster is, the memory containing at least a data structure (see Figure 2 –item 210A-D; paragraph [0038]). The data

structure comprising a list defining membership to a group (see paragraph [0038]; wherein the list is replicated to each job having membership to the group (see paragraph [0046]) and wherein each list is accessed upon each request from a requesting member job to join the group, wherein the request is granted if the other jobs of the group determine that the requesting member job is indicated in each respective list of the other jobs (see Figure 7; paragraph [0054]-[0056]).



### **Grounds of Rejection to be Reviewed on Appeal**

1. Claims 8-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Badovinatz et al.* (US 6,026,426, hereinafter, "*Badovinatz*"), and further in view of *Elley et al.* (US 6,883,100, hereinafter, "*Elley*").

## **ARGUMENTS**

### **Obviousness of Claims 8-23 over *Badovinat* in view of *Elley***

Claims 8-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Badovinat* and further in view of *Elley*. Applicants respectfully traverse this rejection.

The Examiner bears the initial burden of establishing a *prima facie* case of obviousness. See MPEP § 2142. To establish a *prima facie* case of obviousness three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one ordinary skill in the art, to modify the reference or to combine the reference teachings. Second, there must be a reasonable expectation of success. Third, the prior art reference (or references when combined) must teach or suggest all the claim limitations. See MPEP § 2143. The present rejection fails to establish at least the third criteria.

For example, the references, even if combined as suggested in the Office Action, do not disclose (i) upon receiving a request to create a group comprising at least two jobs: creating, on a respective node on which each respective job is running, *a respective list indicating each of the at least two jobs*, or (ii) upon receiving a request to join the group from a requesting member job having membership to the group: *accessing each respective list of each job of the group to determine whether the requesting member job is included in the respective list*, as recited in independent claim 8.

In rejecting claim 8, the Examiner cites to Figure 4 of *Badovinat*, stating “where processing node 1 and processing node 2 each have respective lists indicating process x (job) running)” (see Final Office Action dated July 25, 2006). Applicants respectfully submit that Examiner has misconstrued *Badovinat* and that Figure 4 of *Badovinat* discloses processor 1, processor 2, processor N, group services, internal layer, external layer and process x (see Figure 2 –items 106, 200, 202, 302, 304). Figure 4 does not disclose receiving a request to create a group comprising at least two jobs, as claimed

in independent claim 8. Similarly, Figure 4 does not disclose a list; it discloses the actual processes being executed (see Figure 4, Processing Node 2).

That Figure 4 depicts the actual processes executing on the node is further supported by the fact that *Badovinat* does not once refer to “PROCESS X” as a list and instead refers to “PROCESS X” as an executing process (see *Badovinat* column 4, line 57-61). Additionally, “PROCESS X” is described, in relation to Figure 2, by *Badovinat*:

A process 202 (FIG. 2) desirous of using the facilities of Group Services is coupled to a group Services daemon 200. See column 4, line 15-19.

Applicants also respectfully submit that not only does *Badovinat* not disclose groups of jobs, *Badovinat*, in fact, discloses groups of processors. *Badovinat* states:

The internal layer of Group Services implements functions on a *per processor group* basis. There may be a plurality of *processor groups* in the network. Each *processor group* (also, referred to as a *metagroup*) includes *one or more processors* having a Group Services daemon executing thereon. The *processors of a particular group* are related in that they are executing related processes (*emphasis added*). See column 4, line 50-56.

With respect to the difference between a job group and a processor group, Applicants respectfully submit that the processor group disclosed in Examiner’s citations is not the same as a job group. Paragraphs 39 and 44 of the specification illustrate the distinction between a processor and a job, and Applicants submit the distinction between a processor and a job is well-known. Paragraph 44 defines a node which “includes one or more system processors.” Paragraph 39 discloses that “two or more nodes may define a cluster.” One or more “groups” may be defined within a cluster, the groups corresponding to “a logical grouping of a member or members.” See Paragraph 39. A member is defined as “a job executing on one or more of the nodes within the cluster. *Id.* However, as noted above, a job is not synonymous with a processor.

Further, *Badovinat* is clear in stating that a process group and a processor group are created when a first process request to join a process group.

If this is the first request to join the particular process group, then the name server informs the Group Services Daemon that it is the group leader, INQUIRY 1204 "FIRST REQUEST TO JOIN?". Thus, the processor creates processor group, as described above, and adds the process to the process group, STEP 120 "ADD PROCESS."

*Badovinat* clearly does not disclose receiving a request to create a group comprising at least two jobs.

In the Final Office Action, the Examiner admits that *Badovinat* fails to disclose: *that the requesting member job has membership to the group and accessing each list of each job of the group to determine whether the requesting member job is included in each list*, as claimed. The Examiner argues, however, that these features are well known in the art and would have been an obvious modification of the system disclosed by *Badovinat*, as evidenced by *Elley*. The Examiner argues that *Elley* discloses a system of managing group membership, wherein a server may look at a membership list to determine if a requesting member has membership (see Fig. 5. and column 5, lines 4-10).

Applicants respectfully submit, however, that *Elley* does not teach *that the requesting member job has membership to the group and accessing each list of each job of the group to determine whether the requesting member job is included in each list*, nor does the Examiner cite to any section that teaches this. In fact, *Elley* is only concerned with security within groups of servers and clients, but is no way directed towards groups of jobs, as claimed.

Accordingly, Applicants submit that claim 8 and its dependents are allowable and withdrawal of this rejection is respectfully requested.

Regarding independent claims 12 and 19, the Examiner states that *Badovinat* discloses "a memory device containing a copy of a first list; wherein each copy of the first list indicates jobs with membership to a first group (see column 6, lines 1-10)." Applicants respectfully submit that the Examiner has again misconstrued *Badovinat*.

The cited portion of *Badovintz* is directed to a membership list of processors. The relevant portion of *Badovintz* states:

According to the *membership list for Processor Group X*, Processor 1 is the next group leader. However, if Processor 1 is inactive, then Processor 3 would be chosen to be the new group leader (FIG. 6b). In accordance with the principles of the present invention, in one example, the membership list is stored in memory of each of the processing nodes of the processor group (*emphasis added*) (see column 5 line 63 through column 6, line 3).

As is clear from the cited portion above, a processor group is not a group of jobs. *Badovintz* clearly does not disclose a list indicating jobs with a membership to a first group, as recited in claims 12 and 19.

Accordingly, Applicants submit that claims 12 and 19, as well as their dependents are allowable and withdrawal of this rejection is respectfully requested.

## CONCLUSION

The Examiner errs in finding that claims 8-23 are unpatentable over *Badovintz et al.* in view of *Elley* under 35 U.S.C. § 103(a). Withdrawal of the rejection and allowance of all claims is respectfully requested.

Respectfully submitted and  
**S-signed pursuant to 37 CFR 1.4,**

/Randol W. Read, Reg. No. 43,876/  
Randol W. Read  
Registration No. 43,876  
Patterson & Sheridan, L.L.P.  
3040 Post Oak Blvd. Suite 1500  
Houston, TX 77056  
Telephone: (713) 623-4844  
Facsimile: (713) 623-4846  
Attorney for Appellants

## CLAIMS APPENDIX

1-7. (Canceled)

8. (Previously Presented) A method of managing membership of jobs in a cluster, the method comprising:

- (i) upon receiving a request to create a group comprising at least two jobs:  
creating, on a respective node on which each respective job is running, a respective list indicating each of the at least two jobs; and
- (ii) upon receiving a request to join the group from a requesting member job having membership to the group:  
accessing each respective list of each job of the group to determine whether the requesting member job is included in the respective list.

9. (Original) The method of claim 8, further comprising:  
determining that the requesting member job is included in at least one list; and  
joining the requesting member job to the group.

10. (Original) The method of claim 8, further comprising, upon receiving a request to leave the group from a requesting member job having membership to the group:  
updating each list of each job of the group to remove the requesting member job from the list.

11. (Previously Presented) The method of claim 8, further comprising, upon receiving a request to add a new job to the group:  
for each current member of the group, updating a respective list to include the new job; and  
for a new node, replicating the list to the new job.

12. (Previously Presented) A computer system, comprising a first plurality of nodes, each node comprising:  
a processor configured to execute at least one job; and

a memory device containing a copy of a first list; wherein each copy of the first list indicates jobs with a membership to a first group and wherein each job is configured to access its respective copy of the first list to determine whether a requesting job of another node may be joined to the first group.

13. (Original) The system of claim 12, further comprising a plurality of interfaces configured for adding jobs to the first group, removing jobs from the first group, and joining returning member jobs to the first group.

14. (Original) The system of claim 12, wherein each job is configured to update its respective copy of the first list to include added members.

15. (Original) The system of claim 12, wherein each job is configured to update its respective copy of the first list to remove dropped members.

16. (Original) The system of claim 12, wherein the requesting job is joined to the first group when the first list contains a reference to a node on which the requesting job is running.

17. (Original) The system of claim 12, further comprising:  
a second plurality of nodes; and  
a copy of a second list stored on each of the second plurality of nodes and associated with a job executing on the each of the second plurality of nodes; wherein each copy of the second list indicates a membership to a second group.

18. (Original) The system of claim 17, wherein the copies of the first list and the copies of the second list are each unique on the system.

19. (Previously Presented) A memory of a node in a cluster, the memory containing at least a data structure, the data structure comprising a list defining membership to a group; wherein the list is replicated to each job having membership to the group and wherein each list is accessed upon each request from a requesting member job to join the group, wherein the request is granted if the other jobs of the



group determine that the requesting member job is indicated in each respective list of the other jobs.

20. (Original) The memory of claim 19, wherein each list is updated to include a new job upon each granted request from the new job to be added to the group.

21. (Original) The memory of claim 19, wherein each list is updated to remove a leaving member job upon each granted request from the leaving member job to be removed to the group.

22. (Original) The memory of claim 19, wherein the list is unique within the cluster.

23. (Original) The memory of claim 19, further comprising a plurality of interfaces comprising a first interface invoked by a request to add a new job to the group, a second interface invoked by a request to rejoin a requesting job to the group, a third interface invoked by a request to remove the requesting member job from the group.

## EVIDENCE APPENDIX

None.

## RELATED PROCEEDINGS APPENDIX

None.